Quiz 1 **CSCE 421**

Name:

UIN (or TAMU Email):

Total 5 points.

Question 1 (2 points, 0.5 each)

Which of the following problems are best suited for Machine Learning? Please give a short answer (1-2 sentences) to each of the following problems.

(a) Detecting spam emails.

Yes, the ML algorithm will learn the spam patterns of the emails.

(b) Predicting the house market price in College Station.

Yes, from the previous data, the ML algorithm will understand whether prices will go up or down.

(c) Predicting if Texas A&M will win the next game against Lamar.

Yes, the ML algorithm can get data against Lamar from previous years, as well as this year's data to predict.

(d) Classifying a number into even or odd.

No, the machine just needs to divide the number by 2, no data are needed for this case, no patterns need to be learned.

Question 2 (3 points)
Assume that you have a set of training, validation, and test data, $\mathbf{X}^{\textbf{train}} \in \mathbb{R}^{D\times N_1}$, $\mathbf{X}^{\textbf{val}} \in \mathbb{R}^{D\times N_2}$, and $\mathbf{X}^{\textbf{test}} \in \mathbb{R}^{D\times N_3}$, respectively, with corresponding label vectors $\mathbf{y}^{\textbf{train}} \in \mathbb{R}^{N_1}$, $\mathbf{y}^{\textbf{train}} \in \mathbb{R}^{N_2}$, $\mathbf{y}^{\textbf{train}} \in \mathbb{R}^{N_3}$. You would like to perform classification using K-Nearest Neighbor (K-NN). You are not sure about the type of distance metric to use and are trying to decide between the Euclidean and Manhattan distance. Please briefly explain how you would decide between the two distances and provide a basic pseudocode to do this.
Hint: You can assume a function *pred*=KNN(X,Y,*dist*), which provides a decision for test samples Y using training data X and distance metric *dist*, and a function *acc*=ComputeAcc(*pred*,*lab*), which computes the classification accuracy between predicted class *pred* and actual labels *lab*.

Hyperparameter tuning based on the validation set.

```
Accuracy = [ ]

for dist-type in { Euclidean, Manhattan } do :

        Accuracy = [Accuracy, KNN ($X^{train}$, $X^{val}$, dist_type)]

if    Accuracy [0] < Accuracy [1]

        prediction = KNN ([$X^{train}$, $X^{val}$], $X^{test}$, Manhattan)

else

        prediction = KNN( [$X^{train}$, $X^{val}$], $X^{test}$, Euclidean)
```